

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

Docket No.: 43889-960

#2
9/24/00
PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of

Shuichi TAKAYAMA, et al.

Serial No.:

Group Art Unit:

Filed: June 29, 2000

Examiner:

For: PROCESSOR



**CLAIM OF PRIORITY AND
TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT**

Assistant Commissioner for Patents
Washington, DC 20231

Sir:

In accordance with the provisions of 35 U.S.C. 119, Applicants hereby claim the priority of:

Japanese Patent Application No. 11-259976,
filed September 14, 1999

cited in the Declaration of the present application. A certified copy is submitted herewith.

Respectfully submitted,

MCDERMOTT, WILL & EMERY

[Signature]
Michael E. Fogarty
Registration No. 36,139

600 13th Street, N.W.
Washington, DC 20005-3096
(202) 756-8000 MEF:dtb
Date: June 29, 2000
Facsimile: (202) 756-8087

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

43889-960
Takayama, et al.
June 29, 2000

McDermott, Will & Emery

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.



出願年月日

Date of Application:

1999年 9月14日

出願番号

Application Number:

平成11年特許願第259976号

出願人

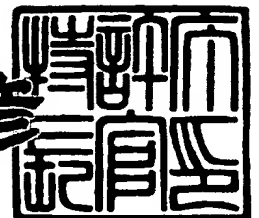
Applicant(s):

松下電器産業株式会社

2000年 5月26日

特許庁長官
Commissioner,
Patent Office

近藤 隆彦



出証番号 出証特2000-3038375

【書類名】 特許願

【整理番号】 5037700091

【提出日】 平成11年 9月14日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/30

【発明者】

 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

 【氏名】 高山 秀一

【発明者】

 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

 【氏名】 檜垣 信生

【発明者】

 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

 【氏名】 鈴木 正人

【特許出願人】

 【識別番号】 000005821

 【氏名又は名称】 松下電器産業株式会社

【代理人】

 【識別番号】 100077931

 【弁理士】

 【氏名又は名称】 前田 弘

【選任した代理人】

 【識別番号】 100094134

 【弁理士】

 【氏名又は名称】 小山 廣毅

【手数料の表示】

【予納台帳番号】 014409

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9601026

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 プロセッサ

【特許請求の範囲】

【請求項 1】 複数のサイズのデータを扱うプロセッサであって、
命令により指定されたオペランドに、前記命令により指定された操作を施すための、前記複数のサイズのうちの最大サイズに対応したビット幅を有する操作手段と、

前記操作手段のうち前記指定されたオペランドのデータサイズに応じた部分のみを動作させるための操作制御手段とを備えたことを特徴とするプロセッサ。

【請求項 2】 請求項 1 記載のプロセッサにおいて、

前記操作手段は、オペランドデータに算術論理演算操作を施すための、前記最大サイズに対応したビット幅を有する算術論理演算手段を備え、

前記算術論理演算手段のうち前記指定されたオペランドのデータサイズに応じた部分のみが動作するように構成されたことを特徴とするプロセッサ。

【請求項 3】 請求項 2 記載のプロセッサにおいて、

前記操作手段は、前記算術論理演算手段を構成する各部分の間で演算情報を伝達するための演算情報伝達手段を更に備え、

前記演算情報伝達手段のうち前記指定されたオペランドのデータサイズに応じた部分のみが動作するように構成されたことを特徴とするプロセッサ。

【請求項 4】 請求項 3 記載のプロセッサにおいて、

前記演算情報伝達手段により伝達される演算情報は、桁上げに関する情報であることを特徴とするプロセッサ。

【請求項 5】 請求項 1 記載のプロセッサにおいて、

前記操作手段は、オペランドデータを保持するための、前記最大サイズに対応したビット幅を有するレジスタ手段を備え、

前記レジスタ手段のうち前記指定されたオペランドのデータサイズに応じた部分のみが動作するように構成されたことを特徴とするプロセッサ。

【請求項 6】 請求項 1 記載のプロセッサにおいて、

前記操作手段は、

オペランドデータを転送するための、前記最大サイズに対応したビット幅を有するバス手段と、

前記バス手段を駆動するための、前記最大サイズに対応したビット幅を有するドライバ手段とを備え、

前記ドライバ手段のうち前記指定されたオペランドのデータサイズに応じた部分のみが動作するように構成されたことを特徴とするプロセッサ。

【請求項 7】 請求項 1 記載のプロセッサにおいて、

前記操作手段は、

オペランドデータを転送するための、前記最大サイズに対応したビット幅を有するバス手段と、

前記バス手段上のオペランドデータを取り込むための、前記最大サイズに対応したビット幅を有するラッチ手段とを備え、

前記ラッチ手段のうち前記指定されたオペランドのデータサイズに応じた部分のみが動作するように構成されたことを特徴とするプロセッサ。

【請求項 8】 請求項 1 記載のプロセッサにおいて、

前記操作手段は、オペランドデータのサイズを拡張するための、前記最大サイズに対応したビット幅を有する拡張手段を備え、

前記拡張手段のうち前記指定されたオペランドのデータサイズに応じた部分のみが動作するように構成されたことを特徴とするプロセッサ。

【請求項 9】 請求項 1 記載のプロセッサにおいて、

前記プロセッサは、RISC型であることを特徴とするプロセッサ。

【請求項 10】 請求項 1 記載のプロセッサにおいて、

前記操作制御手段は、与えられた命令を解釈することにより、該命令により指定されたオペランドについて、該指定されたオペランドのデータサイズに関するサイズ情報を得て、前記操作手段のうち前記サイズ情報により指定された部分のみを動作させる機能を更に有することを特徴とするプロセッサ。

【請求項 11】 請求項 10 記載のプロセッサにおいて、

前記操作手段は、オペランドデータと、該オペランドデータのサイズに関するサイズ情報とを保持するためのレジスタ手段を備え、

前記操作制御手段は、

前記レジスタ手段への特定サイズオペランドデータの設定を要求する第 1 の命令が解読されて、該第 1 の命令により指定されたオペランドのデータサイズに関するサイズ情報が得られたときには、前記オペランドデータとともに前記サイズ情報を前記レジスタ手段に保持させる機能と、

前記レジスタ手段に保持されたオペランドデータの参照を要求する第 2 の命令が解読されたときには、前記レジスタ手段から前記オペランドデータとともに該オペランドデータのサイズに関するサイズ情報を読み出して、前記操作手段のうち前記レジスタ手段から読み出したサイズ情報により指定された部分のみを動作させる機能とを更に有することを特徴とするプロセッサ。

【請求項 1 2】 請求項 1 1 記載のプロセッサにおいて、

前記第 1 の命令は、メモリから前記レジスタ手段へのオペランドデータのロードを要求する命令、又は、前記レジスタ手段への即値オペランドデータの転送を要求する命令であることを特徴とするプロセッサ。

【請求項 1 3】 請求項 1 1 記載のプロセッサにおいて、

前記第 2 の命令は、オペランドのデータサイズに関する指定情報を含まないオペコードフィールドを有する命令であることを特徴とするプロセッサ。

【請求項 1 4】 請求項 1 2 記載のプロセッサにおいて、

前記第 2 の命令は、前記レジスタ手段を参照する算術論理演算命令、又は、前記レジスタ手段からメモリへのオペランドデータのストアを要求する命令であることを特徴とするプロセッサ。

【請求項 1 5】 請求項 1 1 記載のプロセッサにおいて、

前記操作制御手段は、特定サイズへのオペランドデータのサイズの変更を要求する第 3 の命令が解読されて、該第 3 の命令により指定されたオペランドのデータサイズに関する新たなサイズ情報が得られたときには、前記レジスタ手段中の前記指定されたオペランドに対応するデータのサイズ情報を変更する機能を更に有することを特徴とするプロセッサ。

【請求項 1 6】 請求項 1 1 記載のプロセッサにおいて、

前記第 1 の命令は、オペランドデータを符号付きデータとして扱うべきか又は

符号なしデータとして扱うべきかを指定する、オペランドの符号の有無に関する符号情報を含むオペコードフィールドを有する命令であり、

前記レジスタ手段は、前記オペランドデータの符号の有無に関する符号情報を保持する機能を更に有し、

前記操作制御手段は、

前記第 1 の命令が解読されて、該第 1 の命令により指定されたオペランドの符号の有無に関する符号情報が得られたときには、前記オペランドデータとともに前記符号情報を前記レジスタ手段に保持させる機能と、

前記レジスタ手段に保持されたオペランドデータの参照を要求する前記第 2 の命令が解読されたときには、前記レジスタ手段から前記オペランドデータとともに該オペランドデータの符号の有無に関する符号情報を読み出して、該読み出した符号情報に従って前記第 2 の命令が実行されるように前記操作手段を制御する機能とを更に有することを特徴とするプロセッサ。

【請求項 1 7】 請求項 1 6 記載のプロセッサにおいて、

前記操作制御手段は、オペランドデータの符号の有無の変更を要求する第 4 の命令が解読されて、該第 4 の命令により指定されたオペランドの符号の有無に関する新たな符号情報が得られたときには、前記レジスタ手段中の前記指定されたオペランドに対応するデータの符号情報を変更する機能を更に有することを特徴とするプロセッサ。

【請求項 1 8】 請求項 1 7 記載のプロセッサにおいて、

前記操作制御手段は、

与えられた命令を解読するための命令解読器と、

前記レジスタ手段への前記サイズ情報及び前記符号情報の書き込みと、前記レジスタ手段からの前記サイズ情報及び前記符号情報の読み出しとを司るコントローラとを備えたことを特徴とするプロセッサ。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、複数のサイズのデータを扱うプロセッサに関するものである。

【0 0 0 2】

【従来の技術】

近年、機器組み込み用途のプロセッサが扱うデータは多様化している。例えば、32ビット、16ビット、8ビットのサイズのデータを扱うことが要求される。一方では、携帯機器へプロセッサを応用する場合、バッテリーの容量に制限があるため、長時間駆動を行うには低消費電力が要求されている。

【0 0 0 3】

32ビットのアーキテクチャを有する従来のRISC (Reduced Instruction Set Computer) 型プロセッサは、データに算術論理演算操作を施すための32ビットのALU (Arithmetic and Logic Unit) と、データを保持するための32ビットのレジスタファイルと、データを転送するための32ビットのバスとを備えたものであった。このプロセッサで8ビットデータを扱う場合には、このデータのサイズが無条件で32ビットに拡張される。具体的には、8ビットデータが符号付きデータであるならば上位24ビットに符号拡張が、8ビットデータが符号なしデータであるならば上位24ビットにゼロ拡張がそれぞれなされる。16ビットデータを扱う場合にも、このデータのサイズが無条件で32ビットに拡張される。このようにして得られた32ビットデータが、32ビットのALUに供給され、32ビットのレジスタファイルに保持され、また32ビットのバスを通して転送されるようになっていた。

【0 0 0 4】

【発明が解決しようとする課題】

上記従来のRISC型プロセッサによれば、2つの8ビットデータをメモリから順次ロードし、これらを加算し、その結果を表す8ビットデータをメモリにストアするという一連の動作を行う場合であっても、32ビットのALUと、32ビットのレジスタファイルと、32ビットのバスとの全てのハードウェア資源が動作していたので、電力が無駄に消費される問題があった。このことは、16ビットデータを扱う場合にも同様であった。

【0 0 0 5】

本発明の目的は、複数のサイズのデータを扱うプロセッサにおける電力消費の

無駄を削減することにある。

【0006】

【課題を解決するための手段】

上記目的を達成するため、本発明は、複数のサイズのデータを扱うプロセッサにおいて、最大サイズに対応したビット幅を有する操作手段のうち、指定されたオペランドのデータサイズに応じた部分のみを動作させることとしたものである。ここにいう操作手段は、命令により指定されたオペランドに、該命令により指定された操作を施すための手段であって、オペランドデータに算術論理演算操作を施すためのALUと、オペランドデータを保持するためのレジスタファイルと、オペランドデータを転送するためのバスと、バスを駆動するためのドライバと、バス上のオペランドデータを取り込むためのラッチと、オペランドデータのサイズを拡張するための拡張器とを含む。例えば、与えられた命令を解読することにより、該命令により指定されたオペランドについて、該指定されたオペランドのデータサイズに関するサイズ情報を得て、前記操作手段のうち当該サイズ情報により指定された部分のみを動作させる。

【0007】

オブジェクト指向の観点によれば、オペランドデータと、該オペランドデータのサイズに関するサイズ情報とをレジスタファイルに保持させるようにするとよい。この場合において、レジスタファイルへの特定サイズのオペランドデータの設定を要求する第1の命令が解読されて、該第1の命令により指定されたオペランドのデータサイズに関するサイズ情報が得られたときには、オペランドデータとともにサイズ情報をレジスタファイルに保持させる。また、レジスタファイルに保持されたオペランドデータの参照を要求する第2の命令が解読されたときには、レジスタファイルからオペランドデータとともに該オペランドデータのサイズに関するサイズ情報を読み出して、前記操作手段のうち当該レジスタファイルから読み出したサイズ情報により指定された部分のみを動作させる。

【0008】

本発明の他の観点によれば、前記第1の命令は、オペランドデータを符号付きデータとして扱うべきか又は符号なしデータとして扱うべきかを指定する命令で

もある。レジスタファイルには、オペランドデータと、該オペランドデータのサイズに関するサイズ情報とに加えて、該オペランドデータの符号の有無に関する符号情報を保持させる。この場合において、前記第 1 の命令が解読されて、該第 1 の命令により指定されたオペランドの符号の有無に関する符号情報が得られたときには、オペランドデータとともに符号情報をレジスタファイルに保持させる。また、レジスタファイルに保持されたオペランドデータの参照を要求する前記第 2 の命令が解読されたときには、レジスタファイルからオペランドデータとともに該オペランドデータの符号の有無に関する符号情報を読み出して、当該読み出した符号情報に従って前記第 2 の命令が実行されるように前記操作手段を制御する。

【0009】

【発明の実施の形態】

図 1 は、本発明に係るプロセッサの構成例を示している。本プロセッサは、32ビットのアーキテクチャを有する RISC 型プロセッサであって、32ビット（ワード）、16ビット（ハーフワード）、8ビット（バイト）のサイズのデータを扱うものである。

【0010】

本プロセッサは、データに算術論理演算操作を施して、その結果を得るとともにフラグを生成するための 32ビットの ALU を備えている。この 32ビット ALU は、4 個の 8ビット ALU 1 a, 1 b, 1 c, 1 d に分割されている。ゲート 1 1 a, 1 1 b, 1 1 c は、これら 4 個の 8ビット ALU 1 a ~ 1 d の間で桁上げ（キャリ）に関する演算情報を選択的に伝達するためのゲートである。4 個の 8ビット ALU 1 a ~ 1 d と 3 個のゲート 1 1 a ~ 1 1 c との全てが動作するとき、リップルキャリ方式の 32ビット ALU が実現する。2 個の 8ビット ALU 1 c, 1 d と 1 個のゲート 1 1 c のみが動作するとき、16ビット ALU が実現する。また、1 個の 8ビット ALU 1 d のみが動作することもある。フラグは、オペランドのデータサイズに応じたビット位置で生成されるようになっている。なお、図 1 中の符号の末尾の a, b, c, d はそれぞれ、32ビットのうちの最上位 8ビット、中上位 8ビット、中下位 8ビット、最下位 8ビットの位置を表

すよう付されている。

【 0 0 1 1 】

本プロセッサは、8本のレジスタR 0 ～ R 7 で構成された3 2ビットのレジスタファイル2を更に備えている。レジスタR 0 ～ R 7 の各々は、それぞれ8ビットデータを保持するためのデータA部（DA部）2 a、データB部（DB部）2 b、データC部（DC部）2 c及びデータD部（DD部）2 dと、データの符号の有無に関する1ビット情報を保持するための符号情報部（SU部）2 eと、データのサイズに関する2ビット情報を保持するためのサイズ情報部（SZ部）2 fとから構成される。あるレジスタのSU部2 eの1ビットの値が“1”であることは、当該レジスタに保持されたデータが符号付きデータとして扱われるべきことを意味する。同ビットの値が“0”であることは、当該レジスタに保持されたデータが符号なしデータとして扱われるべきことを意味する。あるレジスタのSZ部2 fの2ビットの値が“1 0”であることは、当該レジスタのDA部2 a、DB部2 b、DC部2 c及びDD部2 dが3 2ビットのデータを保持していることを意味する。同2ビットの値が“0 1”であることは、当該レジスタの中のDC部2 c及びDD部2 dのみが1 6ビットのデータを保持していることを意味する。また、同2ビットの値が“0 0”であることは、当該レジスタの中のDD部2 dのみが8ビットのデータを保持していることを意味するようになっている。

【 0 0 1 2 】

本プロセッサは、各々データを転送するための、各々3 2ビット幅を有するAバス3 1、Bバス3 2及びCバス3 3を更に備えている。図1において、2 0は命令列を格納するためのROM（Read-Only Memory）、2 1はROM 2 0から読み出された命令を保持するための命令レジスタ（IR：Instruction Register）、2 4はロードストアユニット、2 5はデータを格納するためのRAM（Random Access Memory）である。

【 0 0 1 3 】

命令レジスタ2 1から得られた即値オペランドデータは、Bバス3 2を駆動するための各々8ビットのドライバ3 a、3 b、3 c、3 dへ供給される。3 2ビ

ットの即値オペランドデータをBバス32へ供給すべき場合には、ドライバ3a～3dが動作する。16ビットの即値オペランドデータをBバス32へ供給すべき場合には、ドライバ3c、3dのみが動作する。また、8ビットの即値オペランドデータをBバス32へ供給すべき場合には、ドライバ3dのみが動作する。

【0014】

RAM25からロードストアユニット24により読み出されたロードデータは、Bバス32を駆動するための各々8ビットのドライバ4a、4b、4c、4dへ供給される。32ビットのロードデータをBバス32へ供給すべき場合には、ドライバ4a～4dが動作する。16ビットのロードデータをBバス32へ供給すべき場合には、ドライバ4c、4dのみが動作する。また、8ビットのロードデータをBバス32へ供給すべき場合には、ドライバ4dのみが動作する。

【0015】

レジスタファイル2から読み出された第1のデータは、Aバス31を駆動するための各々8ビットのドライバ5a、5b、5c、5dへ供給される。DA部2a、DB部2b、DC部2c及びDD部2dから読み出された32ビットのデータをAバス31へ供給すべき場合には、ドライバ5a～5dが動作する。DC部2c及びDD部2dから読み出された16ビットのデータをAバス31へ供給すべき場合には、ドライバ5c、5dのみが動作する。また、DD部2dから読み出された8ビットのデータをAバス31へ供給すべき場合には、ドライバ5dのみが動作する。

【0016】

レジスタファイル2から読み出された第2のデータは、Bバス32を駆動するための各々8ビットのドライバ6a、6b、6c、6dへ供給される。DA部2a、DB部2b、DC部2c及びDD部2dから読み出された32ビットのデータをBバス32へ供給すべき場合には、ドライバ6a～6dが動作する。DC部2c及びDD部2dから読み出された16ビットのデータをBバス32へ供給すべき場合には、ドライバ6c、6dのみが動作する。また、DD部2dから読み出された8ビットのデータをBバス32へ供給すべき場合には、ドライバ6dのみが動作する。

【 0 0 1 7 】

図 1 中の 7 a, 7 b, 7 c 及び 7 d は、各々 A バス 3 1 上のデータを取り込むための、各々 8 ビットのラッチ (D A 1, D B 1, D C 1 及び D D 1) である。取り込むべきデータが 3 2 ビットのサイズを有する場合には、ラッチ 7 a ~ 7 d が動作する。取り込むべきデータが 1 6 ビットのサイズを有する場合には、ラッチ 7 c, 7 d のみが動作する。また、取り込むべきデータが 8 ビットのサイズを有する場合には、ラッチ 7 d のみが動作する。

【 0 0 1 8 】

図 1 中の 8 a, 8 b, 8 c 及び 8 d は、各々 B バス 3 2 上のデータを取り込むための、各々 8 ビットのラッチ (D A 2, D B 2, D C 2 及び D D 2) である。取り込むべきデータが 3 2 ビットのサイズを有する場合には、ラッチ 8 a ~ 8 d が動作する。取り込むべきデータが 1 6 ビットのサイズを有する場合には、ラッチ 8 c, 8 d のみが動作する。また、取り込むべきデータが 8 ビットのサイズを有する場合には、ラッチ 8 d のみが動作する。

【 0 0 1 9 】

本プロセッサは、ラッチ 7 a ~ 7 d に取り込まれたデータのサイズを拡張して、その結果を表すデータを A L U 1 a ~ 1 d の左入力へ供給するための 3 2 ビットの拡張器を更に備えている。この 3 2 ビット拡張器は、4 個の 8 ビット拡張器 (E A 1, E B 1, E C 1 及び E D 1) 9 a, 9 b, 9 c, 9 d に分割されている。4 個の 8 ビット拡張器 9 a ~ 9 d の全てが動作するとき、8 ビット又は 1 6 ビットデータのサイズを 3 2 ビットへ拡張するための拡張器が実現する。2 個の 8 ビット拡張器 9 c, 9 d のみが動作するとき、8 ビットデータのサイズを 1 6 ビットへ拡張するための拡張器が実現する。サイズ拡張されるべきデータが符号付きデータであるならば符号拡張が、サイズ拡張されるべきデータが符号なしデータであるならばゼロ拡張がそれぞれなされる。

【 0 0 2 0 】

本プロセッサは、ラッチ 8 a ~ 8 d に取り込まれたデータのサイズを拡張して、その結果を表すデータを A L U 1 a ~ 1 d の右入力へ供給するための 3 2 ビットの拡張器を更に備えている。この 3 2 ビット拡張器は、4 個の 8 ビット拡張器

(E A 2, E B 2, E C 2 及び E D 2) 10 a, 10 b, 10 c, 10 d に分割されている。4 個の 8 ビット拡張器 10 a ~ 10 d の全てが動作するとき、8 ビット又は 16 ビットデータのサイズを 32 ビットへ拡張するための拡張器が実現する。2 個の 8 ビット拡張器 10 c, 10 d のみが動作するとき、8 ビットデータのサイズを 16 ビットへ拡張するための拡張器が実現する。サイズ拡張されるべきデータが符号付きデータであるならば符号拡張が、サイズ拡張されるべきデータが符号なしデータであるならばゼロ拡張がそれぞれなされる。

【0021】

A L U 1 a ~ 1 d の演算結果を表すデータは、C バス 33 を駆動するための各々 8 ビットのドライバ 12 a, 12 b, 12 c, 12 d へ供給される。32 ビットのデータを C バス 33 へ供給すべき場合には、ドライバ 12 a ~ 12 d が動作する。16 ビットのデータを C バス 33 へ供給すべき場合には、ドライバ 12 c, 12 d のみが動作する。また、8 ビットのデータを C バス 33 へ供給すべき場合には、ドライバ 12 d のみが動作する。レジスタファイル 2 の中に保持すべきデータは、C バス 33 から供給される。また、R A M 25 ヘストアすべきデータは、C バス 33 からロードストアユニット 24 へ供給される。

【0022】

本プロセッサは、命令レジスタ 21 から与えられた命令を解読するための命令解読器 (I D : Instruction Decoder) 22 と、S U 部 2 a 及び S Z 部 2 f の各々の情報の書き込み及び読み出しを司るコントローラ 23 とを更に備えている。命令解読器 22 は、ドライバ 3 a ~ 3 d, 4 a ~ 4 d の駆動のオン・オフを含め、図中の操作手段を構成する種々の要素を制御する。コントローラ 23 も、S U 部 2 a 及び S Z 部 2 f の各々の情報に応じて、図中の操作手段を構成する種々の要素の動作ビット幅等を制御する。

【0023】

図 2 は、本プロセッサの命令セットの一部を示している。本プロセッサは、ロード命令群、サイズ変更命令群、符号有無変更命令群、即値転送命令群、算術論理演算命令群、ストア命令群、ブランチ命令群等を実行できるように構成されている。ただし、図 2 では、ブランチ命令群等の図示が省略されている。図 2 中の

R_n及びR_mは、各々レジスタR₀～R₇のいずれかを意味する。レジスタR₀～R₇の各々はプログラマに対して32ビットレジスタとして定義され、SU部2eとSZ部2fとはプログラマからは見えない。

【0024】

(1) ロード命令群

ロード命令群には、以下の6種類の命令が含まれる。個々の命令のオペコードフィールド (operation code field) は、オペランドの符号の有無に関する指定情報と、オペランドのデータサイズに関する指定情報とを含んでいる。いずれの命令でも、プロセッサの動作制御は命令解読器22によって行われる。

【0025】

(1. 1) l d b mem, R_n

この命令は、メモリ (mem) から8ビットデータをロードして、これを符号付きデータとしてレジスタR_nに設定することを要求する命令である。この命令が解読されると、ロードストアユニット24がRAM25から8ビットのオペランドデータを読み出す。読み出されたデータは、ドライバ4d、Bバス32の最下位8ビット部分を順次経てラッチ8dに取り込まれ、拡張器10d及びALU1dを素通りしてドライバ12dに供給され、Cバス33の最下位8ビット部分を経てレジスタR_nのDD部2dに格納される。これと並行して、レジスタR_nのSU部2eには符号付きデータの保持を表す値“1”が、そのSZ部2fには8ビットデータの保持を表す値“00”がそれぞれ格納される。他のドライバは動作せず、他のラッチやレジスタはデータの保持を続ける。

【0026】

(1. 2) l d b u mem, R_n

この命令は、メモリから8ビットデータをロードして、これを符号なしデータとしてレジスタR_nに設定することを要求する命令である。この命令が解読された場合のプロセッサの動作は、レジスタR_nのSU部2eに符号なしデータの保持を表す値“0”が格納される点を除いて、上記(1. 1)と同じである。

【0027】

(1. 3) l d h mem, R_n

この命令は、メモリから16ビットデータをロードして、これを符号付きデータとしてレジスタR_nに設定することを要求する命令である。この命令が解読されると、ロードストアユニット24がRAM25から16ビットのオペランドデータを読み出す。読み出されたデータは、ドライバ4c, 4d、Bバス32の最下位16ビット部分を順次経てラッチ8c, 8dに取り込まれ、拡張器10c, 10d及びALU1c, 1dを素通りしてドライバ12c, 12dに供給され、Cバス33の最下位16ビット部分を経てレジスタR_nのDC部2c及びDD部2dに格納される。これと並行して、レジスタR_nのSU部2eには符号付きデータの保持を表す値“1”が、そのSZ部2fには16ビットデータの保持を表す値“01”がそれぞれ格納される。他のドライバは動作せず、他のラッチやレジスタはデータの保持を続ける。

【0028】

(1. 4) ldhu mem, R_n

この命令は、メモリから16ビットデータをロードして、これを符号なしデータとしてレジスタR_nに設定することを要求する命令である。この命令が解読された場合のプロセッサの動作は、レジスタR_nのSU部2eに符号なしデータの保持を表す値“0”が格納される点を除いて、上記(1. 3)と同じである。

【0029】

(1. 5) ld mem, R_n

この命令は、メモリから32ビットデータをロードして、これを符号付きデータとしてレジスタR_nに設定することを要求する命令である。この命令が解読されると、ロードストアユニット24がRAM25から32ビットのオペランドデータを読み出す。読み出されたデータは、ドライバ4a~4d、Bバス32の全部を順次経てラッチ8a~8dに取り込まれ、拡張器10a~10d及びALU1a~1dを素通りしてドライバ12a~12dに供給され、Cバス33の全部を経てレジスタR_nのDA部2a、DB部2b、DC部2c及びDD部2dに格納される。これと並行して、レジスタR_nのSU部2eには符号付きデータの保持を表す値“1”が、そのSZ部2fには32ビットデータの保持を表す値“10”がそれぞれ格納される。他のドライバは動作せず、他のラッチやレジスタは

データの保持を続ける。

【0 0 3 0】

(1. 6) l d u m e m, R n

この命令は、メモリから32ビットデータをロードして、これを符号なしデータとしてレジスタR nに設定することを要求する命令である。この命令が解読された場合のプロセッサの動作は、レジスタR nのS U部2 eに符号なしデータの保持を表す値“0”が格納される点を除いて、上記(1. 5)と同じである。

【0 0 3 1】

(2) サイズ変更命令群

サイズ変更命令群には、以下の3種類の命令が含まれる。個々の命令のオペコードフィールドは、オペランドのデータサイズに関する指定情報を含むが、オペランドの符号の有無に関する指定情報を含まない。いずれの命令でも、プロセッサの動作制御は命令解読器22とコントローラ23とによって行われる。

【0 0 3 2】

(2. 1) e x t b R n

この命令は、レジスタR nに保持されたデータ(R nデータ)のサイズを8ビットに変更することを要求する命令である。この命令が解読されると、レジスタR nのS U部2 e及びS Z部2 fの各々の値がコントローラ23に送られる。以降の動作は、S Z部2 fの値、すなわちR nデータのサイズに応じて異なる。

【0 0 3 3】

(i) R nデータが8ビットデータである場合には、何の動作も行われず、全てのドライバは動作せず、全てのラッチやレジスタはデータの保持を続ける。

【0 0 3 4】

(ii) R nデータが16ビットデータである場合には、レジスタR nのS Z部2 fの値が、16ビットデータの保持を表す値“0 1”から、8ビットデータの保持を表す値“0 0”に変更される。ただし、レジスタR nのS U部2 eの値は変更されない。これにより、レジスタR nに保持された16ビットデータは、符号の有無をそのままに、その上位8ビットが無効化される。全てのドライバは動作せず、全てのラッチや他のレジスタはデータの保持を続ける。

【 0 0 3 5 】

(iii) R_nデータが32ビットデータである場合には、レジスタR_nのSZ部2fの値が、32ビットデータの保持を表す値“10”から、8ビットデータの保持を表す値“00”に変更される。ただし、レジスタR_nのSU部2eの値は変更されない。これにより、レジスタR_nに保持された32ビットデータは、符号の有無をそのままに、その上位24ビットが無効化される。全てのドライバは動作せず、全てのラッチや他のレジスタはデータの保持を続ける。

【 0 0 3 6 】

(2. 2) e x t h R_n

この命令は、R_nデータのサイズを16ビットに変更することを要求する命令である。この命令が解釈されると、レジスタR_nのSU部2e及びSZ部2fの各々の値がコントローラ23に送られる。以降の動作は、SU部2e及びSZ部2fの値、すなわちR_nデータの符号の有無とサイズとに応じて異なる。

【 0 0 3 7 】

(i) R_nデータが8ビットデータである場合には、データのサイズ拡張が行われる。具体的には、レジスタR_nのDD部2dから8ビットのオペランドデータが読み出され、これがドライバ6d、Bバス32の最下位8ビット部分を順次経てラッチ8dに取り込まれる。そして、ラッチ8dに取り込まれた8ビットデータのサイズが拡張器10c、10dにより16ビットに拡張される。詳細には、レジスタR_nに保持された8ビットデータが符号付きデータであるならば上位8ビットに符号拡張が、レジスタR_nに保持された8ビットデータが符号なしデータであるならば上位8ビットにゼロ拡張がそれぞれなされる。このサイズ拡張により得られた16ビットデータは、ALU1c、1dを素通りしてドライバ12c、12dに供給され、Cバス33の最下位16ビット部分を経てレジスタR_nのDC部2c及びDD部2dに格納される。これと並行して、レジスタR_nのSZ部2fの値は、8ビットデータの保持を表す値“00”から、16ビットデータの保持を表す値“01”に変更される。ただし、レジスタR_nのSU部2eの値は変更されない。これにより、レジスタR_nに保持されたデータの符号の有無をそのままに、該データのサイズが8ビットから16ビットへ拡張される。他の

ドライバは動作せず、他のラッチやレジスタはデータの保持を続ける。

【0038】

(ii) R_n データが 16 ビットデータである場合には、何の動作も行われない。
。全てのドライバは動作せず、全てのラッチやレジスタはデータの保持を続ける。
。

【0039】

(iii) R_n データが 32 ビットデータである場合には、レジスタ R_n の SZ 部 2 f の値が、32 ビットデータの保持を表す値 “10” から、16 ビットデータの保持を表す値 “01” に変更される。ただし、レジスタ R_n の SU 部 2 e の値は変更されない。これにより、レジスタ R_n に保持された 32 ビットデータは、符号の有無をそのままに、その上位 16 ビットが無効化される。全てのドライバは動作せず、全てのラッチや他のレジスタはデータの保持を続ける。

【0040】

(2. 3) ext R_n

この命令は、 R_n データのサイズを 32 ビットに変更することを要求する命令である。この命令が解読されると、レジスタ R_n の SU 部 2 e 及び SZ 部 2 f の各々の値がコントローラ 23 に送られる。以降の動作は、SU 部 2 e 及び SZ 部 2 f の値、すなわちレジスタ R_n の保持データの符号の有無とサイズとに応じて異なる。

【0041】

(i) R_n データが 8 ビットデータである場合には、データのサイズ拡張が行われる。具体的には、レジスタ R_n の DD 部 2 d から 8 ビットのオペランドデータが読み出され、これがドライバ 6 d、B バス 32 の最下位 8 ビット部分を順次経てラッチ 8 d に取り込まれる。そして、ラッチ 8 d に取り込まれた 8 ビットデータのサイズが拡張器 10 a ~ 10 d により 32 ビットに拡張される。詳細には、レジスタ R_n に保持された 8 ビットデータが符号付きデータであるならば上位 24 ビットに符号拡張が、レジスタ R_n に保持された 8 ビットデータが符号なしデータであるならば上位 24 ビットにゼロ拡張がそれぞれなされる。このサイズ拡張により得られた 32 ビットデータは、ALU 1 a ~ 1 d を素通りしてドライバ

12a ~ 12d に供給され、Cバス33の全部を経てレジスタR_nのDA部2a、DB部2b、DC部2c及びDD部2dに格納される。これと並行して、レジスタR_nのSZ部2fの値は、8ビットデータの保持を表す値“00”から、32ビットデータの保持を表す値“10”に変更される。ただし、レジスタR_nのSU部2eの値は変更されない。これにより、レジスタR_nに保持されたデータの符号の有無をそのままに、該データのサイズが8ビットから32ビットへ拡張される。他のドライバは動作せず、他のラッチやレジスタはデータの保持を続ける。

【0042】

(ii) R_nデータが16ビットデータである場合には、データのサイズ拡張が行われる。具体的には、レジスタR_nのDC部2c及びDD部2dから16ビットのオペランドデータが読み出され、これがドライバ6c、6d、Bバス32の最下位16ビット部分を順次経てラッチ8c、8dに取り込まれる。そして、ラッチ8c、8dに取り込まれた16ビットデータのサイズが拡張器10a ~ 10dにより32ビットに拡張される。プロセッサの以降の動作は、R_nデータが8ビットデータである場合に関する上記(i)と同じである。

【0043】

(iii) R_nデータが32ビットデータである場合には、何の動作も行われぬ。全てのドライバは動作せず、全てのラッチやレジスタはデータの保持を続ける。

【0044】

(3) 符号有無変更命令群

符号有無変更命令群には、以下の2種類の命令が含まれる。個々の命令のオペコードフィールドは、オペランドの符号の有無に関する指定情報を含むが、オペランドのデータサイズに関する指定情報を含まない。いずれの命令でも、プロセッサの動作制御は命令解読器22によって行われる。

【0045】

(3.1) sgn R_n

この命令は、レジスタR_nに保持されたデータ(R_nデータ)を符号付きデー

タに変更することを要求する命令である。この命令が解釈されると、レジスタ R_n の SU 部 2 e に符号付きデータの保持を表す値 “1” が格納される。レジスタ R_n の SZ 部 2 f の値は変更されない。これにより、レジスタ R_n に符号なしデータが保持されていた場合には、そのデータサイズをそのままに、以降は該データが符号付きデータとして扱われることとなる。全てのドライバは動作せず、全てのラッチや他のレジスタはデータの保持を続ける。

【0046】

(3. 2) `unsgn R_n`

この命令は、 R_n データを符号なしデータに変更することを要求する命令である。この命令が解釈されると、レジスタ R_n の SU 部 2 e に符号なしデータの保持を表す値 “0” が格納される。レジスタ R_n の SZ 部 2 f の値は変更されない。これにより、レジスタ R_n に符号付きデータが保持されていた場合には、そのデータサイズをそのままに、以降は該データが符号なしデータとして扱われることとなる。全てのドライバは動作せず、全てのラッチや他のレジスタはデータの保持を続ける。

【0047】

(4) 即値転送命令群

即値転送命令群には、以下の 6 種類の命令が含まれる。個々の命令のオペコードフィールドは、オペランドの符号の有無に関する指定情報と、オペランドのデータサイズに関する指定情報とを含んでいる。いずれの命令でも、プロセッサの動作制御は命令解読器 22 によって行われる。

【0048】

(4. 1) `movb imm8, R_n`

この命令は、オペランドフィールドで指定された 8 ビット即値データ (`imm8`) を符号付きデータとしてレジスタ R_n に転送することを要求する命令である。この命令が解釈されると、命令レジスタ 21 から 8 ビット即値データが読み出される。読み出されたデータは、ドライバ 3 d、B バス 32 の最下位 8 ビット部分を順次経てラッチ 8 d に取り込まれ、拡張器 10 d 及び ALU 1 d を素通りしてドライバ 12 d に供給され、C バス 33 の最下位 8 ビット部分を経てレジスタ

R_nのDD部2dに格納される。これと並行して、レジスタR_nのSU部2eには符号付きデータの保持を表す値“1”が、そのSZ部2fには8ビットデータの保持を表す値“00”がそれぞれ格納される。他のドライバは動作せず、他のラッチやレジスタはデータの保持を続ける。

【0049】

(4. 2) movbu imm8, R_n

この命令は、オペランドフィールドで指定された8ビット即値データを符号なしデータとしてレジスタR_nに転送することを要求する命令である。この命令が解読された場合のプロセッサの動作は、レジスタR_nのSU部2eに符号なしデータの保持を表す値“0”が格納される点を除いて、上記(4. 1)と同じである。

【0050】

(4. 3) movh imm16, R_n

この命令は、オペランドフィールドで指定された16ビット即値データ(imm16)を符号付きデータとしてレジスタR_nに転送することを要求する命令である。この命令が解読されると、命令レジスタ21から16ビット即値データが読み出される。読み出されたデータは、ドライバ3c, 3d、Bバス32の最下位16ビット部分を順次経てラッチ8c, 8dに取り込まれ、拡張器10c, 10d及びALU1c, 1dを素通りしてドライバ12c, 12dに供給され、Cバス33の最下位16ビット部分を経てレジスタR_nのDC部2c及びDD部2dに格納される。これと並行して、レジスタR_nのSU部2eには符号付きデータの保持を表す値“1”が、そのSZ部2fには16ビットデータの保持を表す値“01”がそれぞれ格納される。他のドライバは動作せず、他のラッチやレジスタはデータの保持を続ける。

【0051】

(4. 4) movhu imm16, R_n

この命令は、オペランドフィールドで指定された16ビット即値データを符号なしデータとしてレジスタR_nに転送することを要求する命令である。この命令が解読された場合のプロセッサの動作は、レジスタR_nのSU部2eに符号なし

データの保持を表す値“0”が格納される点を除いて、上記(4. 3)と同じである。

【0052】

(4. 5) `mov imm32, Rn`

この命令は、オペランドフィールドで指定された32ビット即値データ(`imm32`)を符号付きデータとしてレジスタ`Rn`に転送することを要求する命令である。この命令が解読されると、命令レジスタ21から32ビット即値データが読み出される。読み出されたデータは、ドライバ3a~3d、Bバス32の全部を順次経てラッチ8a~8dに取り込まれ、拡張器10a~10d及びALU1a~1dを素通りしてドライバ12a~12dに供給され、Cバス33の全部を経てレジスタ`Rn`のDA部2a、DB部2b、DC部2c及びDD部2dに格納される。これと並行して、レジスタ`Rn`のSU部2eには符号付きデータの保持を表す値“1”が、そのSZ部2fには32ビットデータの保持を表す値“10”がそれぞれ格納される。他のドライバは動作せず、他のラッチやレジスタはデータの保持を続ける。

【0053】

(4. 6) `movu imm32, Rn`

この命令は、オペランドフィールドで指定された32ビット即値データを符号なしデータとしてレジスタ`Rn`に転送することを要求する命令である。この命令が解読された場合のプロセッサの動作は、レジスタ`Rn`のSU部2eに符号なしデータの保持を表す値“0”が格納される点を除いて、上記(4. 5)と同じである。

【0054】

(5) 算術論理演算命令群

算術論理演算命令群には、以下の5種類のレジスタ間演算命令が含まれる。個々の命令のオペコードフィールドは、オペランドのデータサイズに関する指定情報を含まず、またオペランドの符号の有無に関する指定情報を含まない。ここでは、2オペランド形式を採用し、演算及びフラグの生成はデスティネーションオペランドのデータサイズに基づくものとする。いずれの命令でも、プロセッサの

動作制御は命令解読器 2 2 とコントローラ 2 3 とによって行われる。

【0 0 5 5】

(5. 1) a d d R m, R n

この命令は、レジスタ R m に保持されたデータ (R m データ) と、レジスタ R n に保持されたデータ (R n データ) とを加算して、その結果をレジスタ R n に格納することを要求する命令である。この命令が解読されると、レジスタ R n の S U 部 2 e 及び S Z 部 2 f の各々の値と、レジスタ R m の S U 部 2 e 及び S Z 部 2 f の各々の値とがコントローラ 2 3 に送られる。

【0 0 5 6】

次に、R m データの転送について説明する。R m データが 8 ビットデータである場合には、レジスタ R m の D D 部 2 d から 8 ビットのオペランドデータが読み出され、これがドライバ 6 d、B バス 3 2 の最下位 8 ビット部分を順次経てラッチ 8 d に取り込まれる。R m データが 1 6 ビットデータである場合には、レジスタ R m の D C 部 2 c 及び D D 部 2 d から 1 6 ビットのオペランドデータが読み出され、これがドライバ 6 c、6 d、B バス 3 2 の最下位 1 6 ビット部分を順次経てラッチ 8 c、8 d に取り込まれる。R m データが 3 2 ビットデータである場合には、レジスタ R m の D A 部 2 a、D B 部 2 b、D C 部 2 c 及び D D 部 2 d から 3 2 ビットのオペランドデータが読み出され、これがドライバ 6 a ~ 6 d、B バス 3 2 の全部を順次経てラッチ 8 a ~ 8 d に取り込まれる。

【0 0 5 7】

次に、R n データの転送について説明する。R n データが 8 ビットデータである場合には、レジスタ R n の D D 部 2 d から 8 ビットのオペランドデータが読み出され、これがドライバ 5 d、A バス 3 1 の最下位 8 ビット部分を順次経てラッチ 7 d に取り込まれる。R n データが 1 6 ビットデータである場合には、レジスタ R n の D C 部 2 c 及び D D 部 2 d から 1 6 ビットのオペランドデータが読み出され、これがドライバ 5 c、5 d、A バス 3 1 の最下位 1 6 ビット部分を順次経てラッチ 7 c、7 d に取り込まれる。R n データが 3 2 ビットデータである場合には、レジスタ R n の D A 部 2 a、D B 部 2 b、D C 部 2 c 及び D D 部 2 d から 3 2 ビットのオペランドデータが読み出され、これがドライバ 5 a ~ 5 d、A バ

ス 3.1 の全部を順次経てラッチ 7 a ~ 7 d に取り込まれる。

【0058】

ラッチ 7 a ~ 7 d に保持された R n データは、拡張器 9 a ~ 9 d を素通りして、ALU 1 a ~ 1 d の左入力に与えられる。ラッチ 8 a ~ 8 d に保持された R m データは、R n データのサイズが R m データのサイズより大きい場合には、拡張器 10 a ~ 10 d により R n データのサイズまで拡張がなされたうえ、ALU 1 a ~ 1 d の右入力に与えられる。そうでない場合には、R m データが拡張器 10 a ~ 10 d を素通りして ALU 1 a ~ 1 d の右入力に与えられる。R m データのサイズ拡張がなされる際、R m データの符号の有無に関する情報に応じて、“1” ならば符号拡張が、“0” ならばゼロ拡張がそれぞれ選択される。

【0059】

ALU 1 a ~ 1 d 及びゲート 11 a ~ 11 c は、R n データのサイズに応じて動作する。R n データが 8 ビットデータである場合には、ALU 1 d のみが動作して、8 ビットの加算結果と、フラグとが得られる。R n データが 16 ビットデータである場合には、ALU 1 c, 1 d 及びゲート 11 c のみが動作して、16 ビットの加算結果と、フラグとが得られる。R n データが 32 ビットデータである場合には、ALU 1 a ~ 1 d 及びゲート 11 a ~ 11 c の全てが動作して、32 ビットの加算結果と、フラグとが得られる。フラグは、R n データのサイズに応じたビット位置において、かつ R n データの符号の有無に関する情報に応じて生成される。

【0060】

レジスタファイル 2 への加算結果の格納も、R n データのサイズに応じてなされる。R n データが 8 ビットデータである場合には、8 ビットの加算結果がドライバ 12 d に供給され、Cバス 33 の最下位 8 ビット部分を経てレジスタ R n の DD 部 2 d に格納される。R n データが 16 ビットデータである場合には、16 ビットの加算結果がドライバ 12 c, 12 d に供給され、Cバス 33 の最下位 16 ビット部分を経てレジスタ R n の DC 部 2 c 及び DD 部 2 d に格納される。R n データが 32 ビットデータである場合には、32 ビットの加算結果がドライバ 12 a ~ 12 d に供給され、Cバス 33 の全部を経てレジスタ R n の DA 部 2 a

、D B部 2 b、D C部 2 c 及びD D部 2 d に格納される。以上のプロセッサ動作の間、他のドライバは動作せず、他のラッチやレジスタはデータの保持を続ける。

【0061】

(5. 2) s u b R m, R n

この命令は、R mデータからR nデータを減算して、その結果をレジスタR nに格納することを要求する命令である。この命令が解読された場合のプロセッサの動作は、A L U 1 a ~ 1 d における演算操作が減算である点を除いて、上記 (5. 1) と同じである。

【0062】

(5. 3) c m p R m, R n

この命令は、R mデータとR nデータとを比較することを要求する命令である。この命令が解読された場合のプロセッサの動作は、A L U 1 a ~ 1 d における演算操作が減算である点と、ドライバ 1 2 a ~ 1 2 d の全てがCバス 3 3 を駆動しない点と、レジスタR nのD A部 2 a、D B部 2 b、D C部 2 c 及びD D部 2 d が更新されない点を除いて、上記 (5. 1) と同じである。

【0063】

(5. 4) a n d R m, R n

この命令は、R mデータとR nデータとの論理積をレジスタR nに格納することを要求する命令である。この命令が解読された場合のプロセッサの動作は、A L U 1 a ~ 1 d における演算操作が論理積である点を除いて、上記 (5. 1) と同じである。

【0064】

(5. 5) o r R m, R n

この命令は、R mデータとR nデータとの論理和をレジスタR nに格納することを要求する命令である。この命令が解読された場合のプロセッサの動作は、A L U 1 a ~ 1 d における演算操作が論理和である点を除いて、上記 (5. 1) と同じである。

【0065】

(6) ストア命令群

ストア命令群には、次の 1 種類の命令が含まれる。命令のオペコードフィールドは、オペランドのデータサイズに関する指定情報を含まず、またオペランドの符号の有無に関する指定情報を含まない。プロセッサの動作制御は命令解読器 2 2 とコントローラ 2 3 とによって行われる。

【0 0 6 6】

(6. 1) s t R n, m e m

この命令は、レジスタ R n に保持されたデータ (R n データ) をメモリにストアすることを要求する命令であって、メモリへのストアは R n データのサイズに基づく。この命令が解読されると、レジスタ R n の S U 部 2 e 及び S Z 部 2 f の各々の値がコントローラ 2 3 に送られる。R n データが 8 ビットデータである場合には、レジスタ R n の D D 部 2 d から 8 ビットのオペランドデータが読み出され、これがドライバ 6 d、B バス 3 2 の最下位 8 ビット部分を順次経てラッチ 8 d に取り込まれ、拡張器 1 0 d 及び A L U 1 d を素通りしてドライバ 1 2 d に供給され、C バス 3 3 の最下位 8 ビット部分を経て R A M 2 5 に格納される。R n データが 1 6 ビットデータである場合には、レジスタ R n の D C 部 2 c 及び D D 部 2 d から 1 6 ビットのオペランドデータが読み出され、これがドライバ 6 c, 6 d、B バス 3 2 の最下位 1 6 ビット部分を順次経てラッチ 8 c, 8 d に取り込まれ、拡張器 1 0 c, 1 0 d 及び A L U 1 c, 1 d を素通りしてドライバ 1 2 c, 1 2 d に供給され、C バス 3 3 の最下位 1 6 ビット部分を経て R A M 2 5 に格納される。R n データが 3 2 ビットデータである場合には、レジスタ R n の D A 部 2 a、D B 部 2 b、D C 部 2 c 及び D D 部 2 d から 3 2 ビットのオペランドデータが読み出され、これがドライバ 6 a ~ 6 d、B バス 3 2 の全部を順次経てラッチ 8 a ~ 8 d に取り込まれ、拡張器 1 0 a ~ 1 0 d 及び A L U 1 a ~ 1 d を素通りしてドライバ 1 2 a ~ 1 2 d に供給され、C バス 3 3 の全部を経て R A M 2 5 に格納される。他のドライバは動作せず、他のラッチやレジスタはデータの保持を続ける。

【0 0 6 7】

以上で図 2 中の全ての命令群を説明した。次に、2 つの 8 ビットデータをメモ

りから順次ロードし、これらを加算し、その結果を表す8ビットデータをメモリにストアするという一連の動作を行うことを要求する命令列、すなわち、

命令1: l d b mem, R 0

命令2: l d b mem, R 1

命令3: a d d R 1, R 0

命令4: s t R 0, mem

が実行される場合の、図1のプロセッサの動作を説明する。まず、命令1の実行により、RAM 25から8ビットデータがロードされ、これが符号付きデータとしてレジスタR0のDD部2dに設定される。レジスタR0のSZ部2fには、8ビットデータの保持を表す値“00”が格納される。次に、命令2の実行により、RAM 25から更に8ビットデータがロードされ、これが符号付きデータとしてレジスタR1のDD部2dに設定される。レジスタR1のSZ部2fには、8ビットデータの保持を表す値“00”が格納される。続いて、命令3の実行により、レジスタR0及びR1の各々のSZ部2fが参照されて、R0データとR1データとの8ビット加算が行われ、その結果を表す8ビットデータがレジスタR0に格納される。この際、R0データは拡張器9dを素通りしてALU1dの左入力に、R1データは拡張器10dを素通りしてALU1dの右入力にそれぞれ与えられる。ALU1dは8ビットの加算結果を生成し、この加算結果はレジスタR0のDD部2dに格納される。最後に、命令4の実行により、レジスタR0のSZ部2fが参照されて、当該レジスタR0に保持された8ビットの加算結果がRAM 25にストアされる。

【0068】

以上の一連のプロセッサ動作では、ALU1a～1c、レジスタR0、R1の各々のDA部2a、DB部2b及びDC部2c、ドライバ4a～4c、5a～5c、6a～6c、ラッチ7a～7c、8a～8c、拡張器9a～9c、10a～10c、ゲート11a～11c、ドライバ12a～12c、並びに、A、B及びCバス31、32、33の各々の上位24ビット部分は決して使用されないので、電力消費の無駄が削減される。上記命令列中の例えば命令1（ロード命令）を即値転送命令に置き換えても同様である。また、図1のプロセッサで16ビット

データを扱う場合にも消費電力削減効果が得られる。

【0069】

しかも、図1のプロセッサによれば、レジスタファイル2を構成する8本のレジスタR0～R7の各々にSZ部2fを設け、RAM25からレジスタファイル2への特定サイズオペランドデータのロードを要求する命令や、レジスタファイル2への特定サイズの即値オペランドデータの転送を要求する命令が解読されたときには、オペランドデータのサイズ情報をSZ部2fに保持させ、レジスタファイル2の中のオペランドデータの参照を要求する算術論理演算命令、又は、レジスタファイル2からRAM25へのオペランドデータのストアを要求する命令が解読されたときには、レジスタファイル2のSZ部2fからサイズ情報を読み出して、操作手段を構成する種々の要素のうち当該サイズ情報により指定された部分のみを動作させることとしたので、命令数を増やさずに、従来のプログラム資産を活用しながら上記消費電力削減効果を得ることができる。

【0070】

なお、本発明は、複数のサイズのデータを扱うプロセッサである限り、RISC型に限らず、例えばCISC (Complex Instruction Set Computer) 型のプロセッサにも適用可能である。その他、例えば次のような変形が可能である。

【0071】

(a) 算術論理演算命令群において、演算及びフラグの生成はデスティネーションオペランドのデータサイズに基づくとしたが、ソースオペランドとデスティネーションオペランドとのうちデータサイズの大きい方に基づくとしてもよい。

【0072】

(b) 算術論理演算命令群は2オペランド形式としたが、3オペランド形式であってもよい。その場合、デスティネーションオペランドのデータサイズは、2つのソースオペランドのデータサイズのうち大きい方に基づくとするのが好適である。

【0073】

(c) レジスタファイル、ラッチ、拡張器、ALU、ドライバ、バスを最上位8ビット、中上位8ビット、中下位8ビット、最下位8ビットの4つの部分に分

けて制御しているが、最上位 8 ビットと中上位 8 ビットとを一体化して、最上位 16 ビット、中位 8 ビット、最下位 8 ビットの 3 つの部分に区切って制御してもよい。

【0074】

(d) データサイズを 8 ビット、16 ビット、32 ビットの 3 種類としているが、それに 24 ビットのデータサイズを加えて 4 種類としてもよい。こうすれば、レジスタファイル、ラッチ、拡張器、ALU、ドライバ、バスを最上位 8 ビット、中上位 8 ビット、中下位 8 ビット、最下位 8 ビットの 4 つの部分に分けて制御する図 1 の構成が活かされる。

【0075】

(e) データサイズを 8 ビット、16 ビット、32 ビット、64 ビットの 4 種類とすることも可能である。例えば、レジスタファイル、ラッチ、拡張器、ALU、ドライバ、バスのビット幅を 64 ビットに拡張、最上位 32 ビット、中上位 16 ビット、中下位 8 ビット、最下位 8 ビットの 4 つの部分に分けて制御するようにすればよい。

【0076】

(f) データサイズは 8 ビットの倍数に限らない。ALU の出力からレジスタファイル 2 の入力に至る経路上のいずれかの位置にオペランドデータの有効サイズ検出機構を設け、検出した有効サイズを SZ 部 2 f に保持させるようにしてもよい。例えば、名目上は 32 ビットの符号なしデータであっても、実際には最上位 28 ビットが全て 0 値ビットであることもある。このような場合には、4 ビットデータの保持を意味するサイズ情報を SZ 部 2 f に保持させ、このサイズ情報に応答して、操作手段を構成する種々の要素の最下位 4 ビット部分のみを動作させればよい。

【0077】

(g) レジスタファイル 2 の中に SU 部 2 e 及び SZ 部 2 f を設けず、算術論理演算命令の実行時にレジスタファイル 2 から 2 つの 32 ビットデータを読み出し、両データからその有効サイズと符号付き／符号なしの区別とを検出するためのサイズ／符号検出部を別に設けるようにしてもよい。このサイズ／符号検出部

は、各データの最上位ビット側からビットサーチを行い、最初に1が立っているビットの位置を検出する。その値の大きい方を有効サイズとし、これをコントローラ23に入力する。また、ビットサーチの結果、最上位ビットに1が立っている場合には「符号付き」の情報を、立っていない場合には「符号なし」の情報をコントローラ23に入力する。この一連の動作によって、コントローラ23は上記と同様の信号を入力されるので、以後の動作は同様である。ただし、ALU等の操作手段は動作幅が1ビット単位で分割されているものとする。なお、動作幅を1ビット単位ではなく例えば4ビット単位で分割することとすれば、サイズ／符号検出部でビットサーチを行わず、代わりに4ビット毎の論理操作をすればよいので効率的である。

【0078】

(h) マルチタスク処理に対応するためには、SU部2e及びSZ部2fをコンテキストブロックに追加し、タスク切り換え時にはこれらを退避又は復帰するようにすればよい。

【0079】

【発明の効果】

以上説明してきたとおり、本発明によれば、複数のサイズのデータを扱うプロセッサにおいて、最大サイズに対応したビット幅を有する操作手段のうち、指定されたオペランドのデータサイズに応じた部分のみを動作させることとしたので、当該プロセッサにおける電力消費の無駄を削減することができる。

【0080】

また、オペランドデータのサイズ情報をレジスタファイルに保持させ、当該オペランドデータの参照を要求する算術論理演算命令等が解読されたときには、レジスタファイルからオペランドデータとともにサイズ情報を読み出して、前記操作手段のうち当該サイズ情報により指定された部分のみを動作させることとすれば、命令数を増やさずに、従来のプログラム資産を活用しながら上記消費電力削減効果を得ることができる。

【図面の簡単な説明】

【図1】

本発明に係るプロセッサの構成例を示すブロック図である。

【図 2】

図 1 のプロセッサの命令セットの一部を示す図である。

【符号の説明】

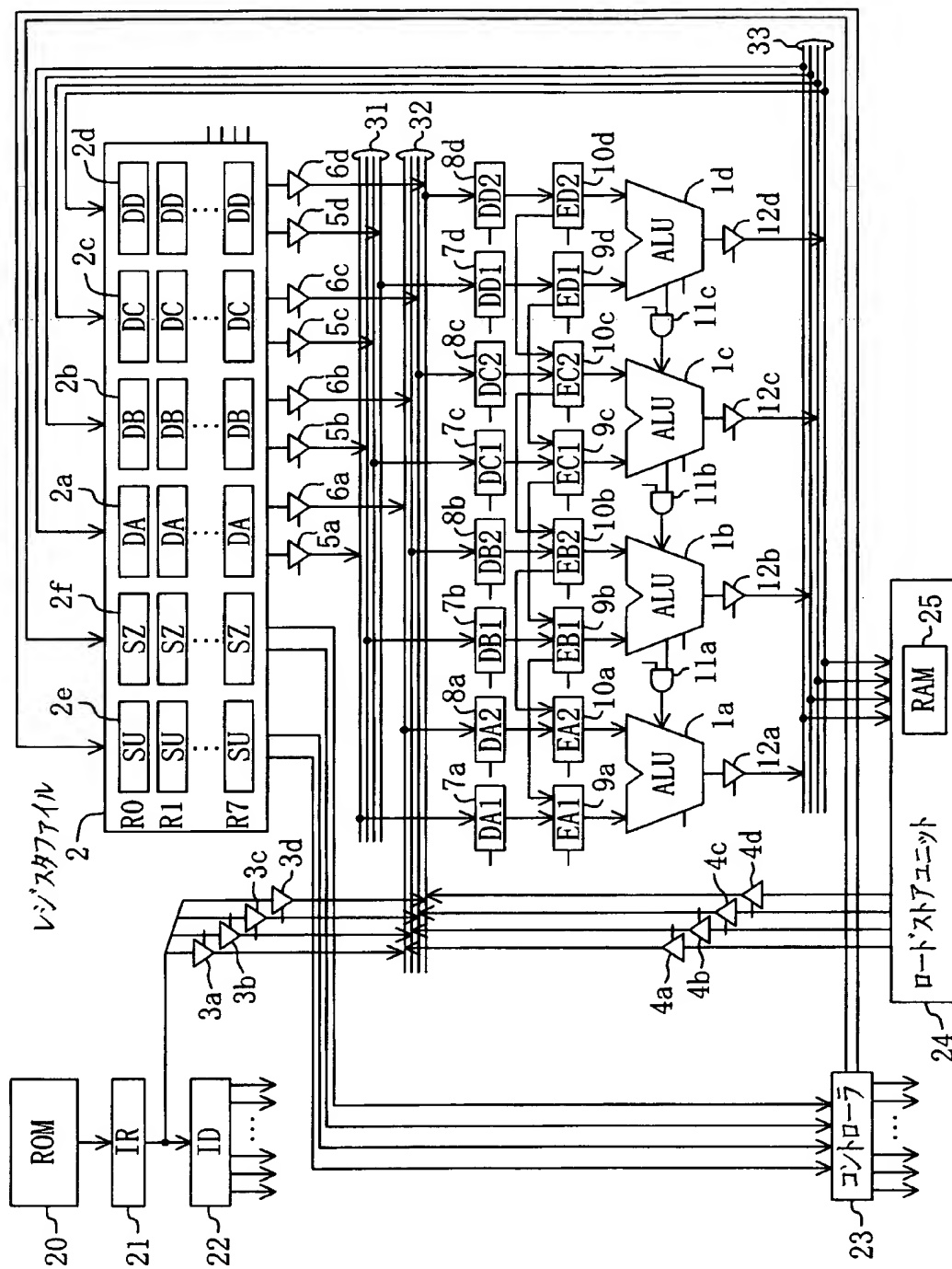
- 1 a ~ 1 d A L U [算術論理演算手段]
- 2 レジスタファイル [レジスタ手段]
- 2 a データ A 部 (D A 部)
- 2 b データ B 部 (D B 部)
- 2 c データ C 部 (D C 部)
- 2 d データ D 部 (D D 部)
- 2 e 符号情報部 (S U 部)
- 2 f サイズ情報部 (S Z 部)
- 3 a ~ 3 d ドライバ [ドライバ手段]
- 4 a ~ 4 d ドライバ [ドライバ手段]
- 5 a ~ 5 d ドライバ [ドライバ手段]
- 6 a ~ 6 d ドライバ [ドライバ手段]
- 7 a ~ 7 d ラッチ [ラッチ手段]
- 8 a ~ 8 d ラッチ [ラッチ手段]
- 9 a ~ 9 d 拡張器 [拡張手段]
- 1 0 a ~ 1 0 d 拡張器 [拡張手段]
- 1 1 a ~ 1 1 c ゲート [演算情報伝達手段]
- 1 2 a ~ 1 2 d ドライバ [ドライバ手段]
- 2 0 R O M
- 2 1 命令レジスタ (I R)
- 2 2 命令解読器 (I D) [操作制御手段]
- 2 3 コントローラ [操作制御手段]
- 2 4 ロードストアユニット
- 2 5 R A M
- 3 1 A バス [バス手段]

3.2 Bバス [バス手段]

3.3 Cバス [バス手段]

【書類名】 図面

【図 1】



【図 2】

(1) ロード命令群

- (1.1) ldb mem, Rn
- (1.2) ldbu mem, Rn
- (1.3) ldh mem, Rn
- (1.4) ldhu mem, Rn
- (1.5) ld mem, Rn
- (1.6) ldu mem, Rn

(2) サイズ変更命令群

- (2.1) extb Rn
- (2.2) exth Rn
- (2.3) ext Rn

(3) 符号有無変更命令群

- (3.1) sgn Rn
- (3.2) unsgn Rn

(4) 即値転送命令群

- (4.1) movb imm8, Rn
- (4.2) movbu imm8, Rn
- (4.3) movh imm16, Rn
- (4.4) movhu imm16, Rn
- (4.5) mov imm32, Rn
- (4.6) movu imm32, Rn

(5) 算術論理演算命令群

- (5.1) add Rm, Rn
- (5.2) sub Rm, Rn
- (5.3) cmp Rm, Rn
- (5.4) and Rm, Rn
- (5.5) or Rm, Rn

(6) ストア命令群

- (6.1) st Rn, mem

【書類名】 要約書

【要約】

【課題】 複数のサイズのデータを扱うプロセッサにおける電力消費の無駄を削減する。

【解決手段】 レジスタファイル 2 を構成する複数本のレジスタ R 0 ～ R 7 の各々に S Z 部 2 f を設け、R A M 2 5 からレジスタファイル 2 への特定サイズのオペランドデータのロードを要求する命令や、レジスタファイル 2 への特定サイズの即値オペランドデータの転送を要求する命令が解読されたときには、オペランドデータのサイズ情報を S Z 部 2 f に保持させ、レジスタファイル 2 中のオペランドデータの参照を要求する算術論理演算命令、又は、レジスタファイル 2 から R A M 2 5 へのオペランドデータのストアを要求する命令が解読されたときには、レジスタファイル 2 の S Z 部 2 f からサイズ情報を読み出して、操作手段を構成する A L U 1 a ～ 1 d 等の種々の要素のうち当該サイズ情報により指定された部分のみを動作させる。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000005821]

1. 変更年月日	1990年 8月28日
[変更理由]	新規登録
住 所	大阪府門真市大字門真1006番地
氏 名	松下電器産業株式会社